

Atividade 1

3- Os valores impressos foram diferentes a cada execução? Por que? Sim. A partir de 2 threads, os valores impressos foram todos diferentes. Isso ocorre porque a estrutura interna do "for" é contida pelas instruções de leitura da variável global, incremento e escrita nela, com isso, são necessárias algumas instruções de máquina. No entanto, a concorrência entre as threads pode realizar as instruções de maneira não determinística, acarretando em até n leituras seguidas do mesmo valor da variável global (rodando com n threads), consequentemente, a escrita nessa variável será o mesmo valor. Logo, no lugar de ter n incrementos resultando em n valores diferentes, nós temos n incrementos que resultam em n valores iguais, então o resultado final não será exatamente $100000 * n$, será um valor entre 100000 (se todas as leituras ocorrerem simultaneamente) e $100000 * n$ (se não ocorrer leituras simultâneas) devido às escritas iguais que eventualmente acontecem.

4- É possível que a variável soma termine com valor acima de 200000 quando executamos com 2 threads? Por que?

Não, porque mesmo que as instruções sejam realizadas de maneira que não ocorra duas leituras de um mesmo valor e escritas iguais (entrelaçamento das instruções), o máximo que a variável "soma" pode receber é 200000, pois a primeira thread incrementará 100000 vezes e a segunda thread também.

Atividade 2

2- Os valores impressos foram sempre o valor esperado? Por que? Sim, porque desta vez uma thread não pode realizar a leitura da variável global antes da escrita realizada por uma outra thread dado que essa já tenha iniciado a leitura. Com isso, sempre serão lidos valores diferentes da variável global e, consequentemente, escritas com valores diferentes.

Atividade 3

2- Quais valores devem ser impressos na saída do programa?

Devem ser impressos valores da variável "soma" que são divisíveis por 10 (não necessariamente todos porque a cada execução do código as threads operam em ordens diferentes).

3- Os valores impressos foram os valores esperados? Por que?

Não, foram impressos alguns valores que não eram divisíveis por 10 (apareceram alguns valores terminando em 1). Isso ocorre justamente pela operação de condição e print da variável. No momento atual, o código não possui um lock isolando essas operações, então pode ocorrer de: ocorre a leitura da variável soma, é constatado que é divisível por 10, entra

no "if". No entanto, antes da operação de print, pode aparecer outra(s) operação(ões) de outra(s) thread(s) que realiza(m) o incremento nessa variável (violação de atomicidade). Quando voltamos à thread extra para realizar o print, a variável acabou de ter sido incrementada, então será impresso o valor atual dela, que infelizmente será diferente do lido durante a operação de condição.

4- Agora quais valores espera-se que sejam impressos na saída do programa?

Usando o lock para isolar a leitura da variável e a impressão dela sem que outra(s) thread(s) apareça(m) para violar a atomicidade e realizar um incremento inesperado, é esperado que só sejam impressos valores divisíveis por 10.

5- O problema foi resolvido? Justifique.

Sim, porque agora as operações de leitura da variável e escrita na tela ocorrem sequencialmente, sem aparecer incrementos da variável realizado por outras threads.